Motorcycle Theft Alarm

with dual sensors, tamper protection and remote control

Design by G. Visschers

This alarm system uses two PIC16F84 processors and is relatively easy to build and install. The cost is very low compared to ready-made systems, while its effectiveness leaves nothing to be desired!

Recently we have seen in the media that there has been a substantial increase in the number of motorcycle and scooter thefts. Large gangs operate together and steal certain makes on order. High up on the list of these criminals are the exclusive models of Ducati and Harley-Davidson, but the more expensive BMWs are also targeted, as is the £20,000 Honda Goldwing. Mind you, it’s not the case that owners of less expensive machines are immune from theft, since many are stolen in that category as well. It is a sad state of affairs, but nevertheless true.

The police have recently had more success in tackling these particular crimes, but for those who care about their expensive two-wheeler a good alarm system remains a necessity.

Unfortunately, a reliable theft

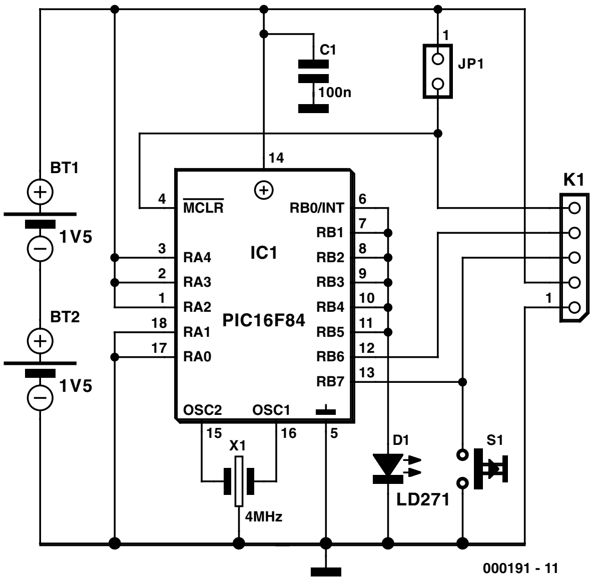

Figure 1. The remote control consists of little more than a (programmed) PIC controller and an IR LED.

alarm is anything but cheap. A decent alarm costs in the region of £250 and on top of that comes the installation charge. When faced with these amounts many people would hesitate to proceed and wonder if there isn't a cheaper alternative.

Of course it is possible to find systems that cost less, but the lower price often reflects a lower reliability. After these deliberations an electronics engineer would soon consider a DIY design. A well-designed alarm needn't be very complex from an electronics point of view, while building and installing it yourself has the advantage that the installation can be adapted optimally for the bike or scooter in question.

The biggest advantage however is the price: it is approximately a quarter of the previously mentioned £250!

Requirements

So what are the requirements of a reliable theft alarm for a bike or scooter?

It should mainly satisfy the following four points:

- The circuit should have a sensitive sensor that reacts to any movement. The infrared and ultrasonic sensors used in house and car alarms are no good for bikes. There really is a need for a type of sensor that can detect movements or vibrations mechanically.
- A conspicuous optical and acoustic alarm signal is obviously a must. This is fortunately not very difficult to implement.
- The alarm should be adequately protected against tampering. This is not so simple and requires more thought.
- And of course only the owner should be able set and clear the alarm with its correct code, preferably with the use of a remote control.

Design overview

What design issues are there when we put the previous wish list into practice and how is it implemented?

To start with we incorporated two different types of movement sensor. That may seem like overkill, but it costs very little and it's better to be safe than sorry. A pair of standard mercury tilt switches detects tilting movements of the bike. On top of that we're using a modified piezo transducer as a vibration sensor; this is particularly effective and can be made so sensitive that it reacts to a slight knock to the bike.

And then there is the tamper protection. This consists of two sections as well. An LDR (light dependent resistor) triggers the alarm when some miscreant removes the buddy seat or opens the compartment where the alarm is mounted. An extra fuse is used to prevent the alarm from malfunctioning when somebody shorts the supply of the indicators.

When the alarm is triggered the indicators of the bike start flashing and a compact but very loud piezo siren goes off. A small infrared transmitter controls the alarm and that obviously needs a matching infrared receiver module to pick up the signal.

At the hart of the circuit is a PIC16F84, containing a simple program. At this point a lot of readers may wonder if this circuit could do without a programmed microcontroller. And of course this would have been possible. But it is a fact that the circuit has to perform many functions: the signals from the sensors have to be interpreted, the IR signal from the remote control has to be detected and then arm or disarm the alarm, it has to generate control signals for the indicators and siren, by law the alarm has to be turned off after sounding for 20 seconds, when it should revert back to its armed state, and finally, the alarm may only go off a maximum of three times after it has been armed. This is a big list and one that requires an elaborate circuit if no microcontroller is used. And besides, there isn't much spare room available in a bike or scooter.

Since the issue of space is even more important for the remote control, we have gratefully made use of the services provided by a PIC16F84, this time in an SMD package. This controller has an easy job, since it only needs to produce the right 36 kHz pulsed signal for the IR transmitter LED.

Both PICs are available ready programmed via our Readers Services, but for those of you who would like to program them themselves we have made the source and HEX files available on our website free of charge under number **000191-11**.

Miniature IR transmitter

By now it's about time to take a closer look at the circuit diagrams. To start with we'll have a look at the remote control, since this circuit contains very few components apart from the small SMD PIC. This circuit is shown in **Figure 1**. Resonator X1 takes care of the required 4 MHz clock signal, D1 is the infrared transmitter LED and S1 functions as the start switch. Connector K7 has been added for the programming of the PIC, since this would otherwise be rather tricky with a SMD package. Before the PIC can be programmed jumper JP1 should be removed.

The operation of the program stored in IC1

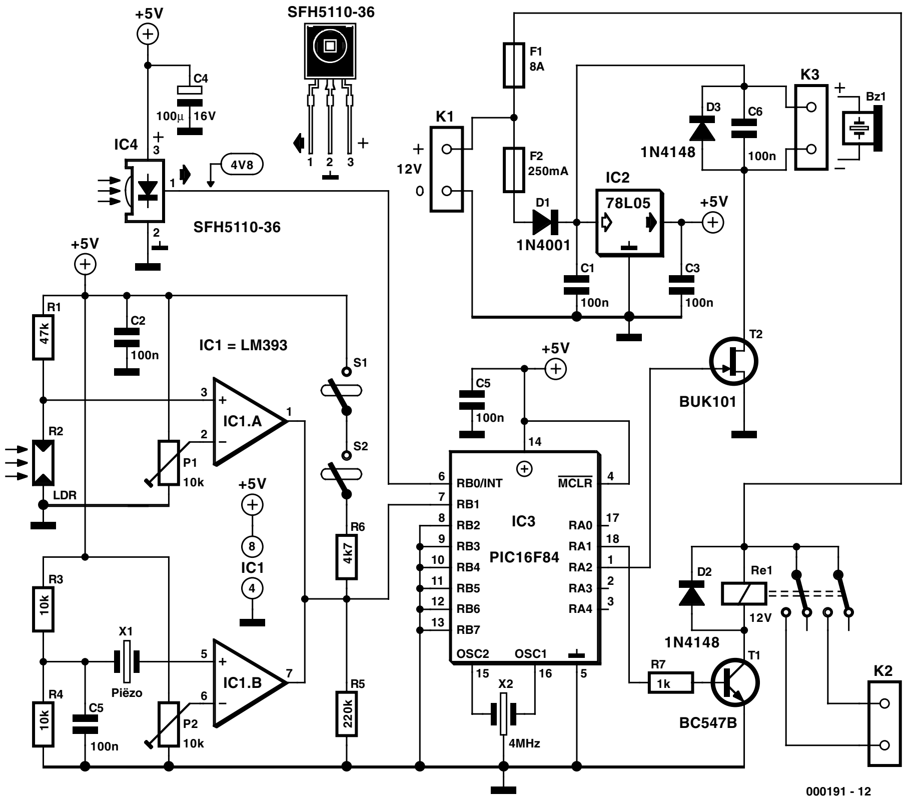

Figure 2. A programmed PIC is also found at the heart of the alarm. On the left are the IR receiver and sensors; on the right are the alarm indicators.

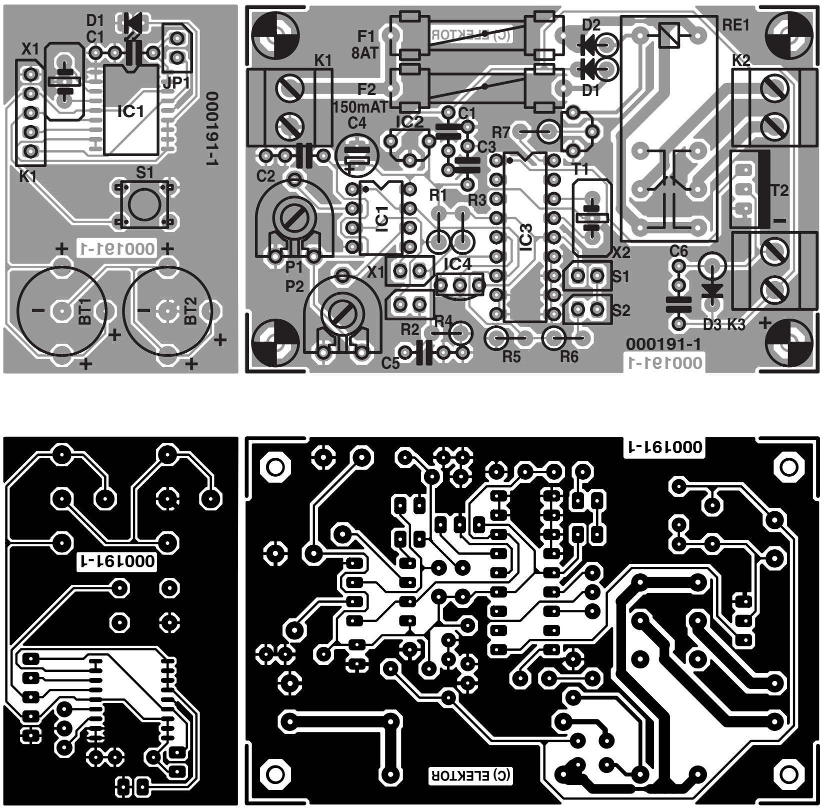

Figure 3. The transmitter and alarm come as one PCB and should be cut apart with a saw.

is very simple. When S1 is pressed the PIC comes out of its sleep-mode, after which it transmits the fixed code as a series of pulses via outputs RB0 to RB5. The PIC then reverts back to its sleep-mode. The IR code is transmitted to the receiver in the alarm by D1. The reason for the parallel connection of RB0 to RB5 is to provide sufficient current to drive the LED.

Although the supply voltage to the PIC16F84 should be at least 5 V according to the datasheets, in practice it appears that it can function reliably down to 2.8 V. For this reason the power is supplied by two 1.5 V button cells connected in series. Since the current consumption of the PIC in sleep-mode is negligible, a pair of button cells should last for at least a year.

COMPONENTS LIST
Transmitter

Capacitors:
C1 = 100nF

Semiconductors:
D1 = IR LED, e.g., CQW13 (Conrad Electronics # 184551-89)
IC1 = PIC16F84-04/SO, programmed, order code **000191-41**

Miscellaneous:
JP1 = 2-way pinheader with jumper
K1 = 5-way pinheader
S1 = miniature pushbutton
X1 = 3-pin 4-MHz ceramic resonator
BT1,BT2 = 1.5V button cell with PCB mount holder 16×1.6 mm (e.g., CR1616)
Enclosure: Teko 11121 (Conrad Electronics # 543287)
PCB: see under ‘alarm’

COMPONENTS LIST
Alarm

Resistors:
R1 = 47kΩ
R2 = LDR
R4,R4 = 10kΩ
R5 = 220kΩ
R6 = 4kΩ7
R7 = 1kΩ

Alarm circuit in detail

The circuit diagram of the actual alarm is shown in **Figure 2**. At first sight it is clear that the use of the PIC controller has kept the circuit nice and simple.

The microcontroller can be found at the hart of the circuit as IC3. Apart from the 4 MHz resonator (X2) that provides the clock signal, there is little more to be said about this because the on-chip software performs all tasks. These tasks have been listed previously. Worth mentioning is that two (unused) outputs can be used for faultfinding: RA0 reflects the status of the alarm (‘0’ when in standby, ‘1’ when armed) and RA3 becomes a ‘1’ when an IR signal is detected.

The integrated IR receiver IC4, which is found in the top-left of the circuit diagram, detects the signal from the transmitter. This should obviously be mounted in such a way that it can ‘see’ the transmitter and it should be connected to the PCB via a short cable. The received signal is passed on to input RB0 of the PIC, which validates the code and toggles the state of the alarm: so ‘standby’ changes to ‘armed’ and the next press of the transmitter button changes it from ‘armed’ to ‘standby’. Two flashes of the indicators indicate each change in status. When the alarm reverts to standby the number of extra flashes indicate how many times the alarm has gone off during the previous armed period.

P1,P2 = 10kΩ

Capacitors:
C1,C2,C3,C5,C6 = 100nF
C4 = 100µF 16V radial

Semiconductors:
D1 = 1N4001
D2,D3 = 1N4148
T1 = BC547
T2 = BUK101
IC1 = LM393
IC2 = 78L05
IC3 = PIC16F84-04/P, programmed, order code **000191-42**
IC4 = SFH5110-36

Miscellaneous:
K1,K2,K3 = 2-way PCB terminal block, lead pitch 5mm
RE1 = 12V relay, 2 make contacts, 8A, (e.g., Schrack RT424012)
S1,S2 = mercury tilt switch (Conrad Electronics # 700452)
BZ1 = 12V siren, e.g., Alecto
X1 = piezo element (Conrad Electronics # 712930)
X2 = 3-pin 4-MHz resonator
F1 = fuse, 8A(T) (slow), with holder
F2 = fuse, 250 mA(T) (slow), with holder
PCB (alarm & transmitter): order code **000191-1** (see Readers Service page)
Disk, IC source code files, order code **000191-11** or Free Download

Below IC4 in the circuit diagram we find the sensor section, consisting of the mercury tilt switches (S1/S2) and two comparators (IC1ab), which take the outputs of the LDR and piezo transducer respectively. The circuit has been designed such that input RB1 becomes low when there is an alarm condition. When the bike is moved, switch S1 and/or S2 becomes open circuit and RB1 is pulled low by R6. When light falls onto LDR R2 or piezo transducer X1 produces a voltage due to vibrations, the open collector output of IC1a or IC1b respectively starts to conduct, also pulling RB1 low. The sensitivity of the LDR can be adjusted with P1 and that of the piezo transducer with P2.

When the alarm is triggered, output RA2 turns FET T2 on, which activates the powerful piezo siren connected to K3. The FET used for T2 is a type that can be driven directly by an output from a TTL compatible IC, and can switch a maximum of 5 A. Since the siren requires only about 140 mA, the FET operates well within its limits. A FET has been chosen here rather than a relay because they are about the same price, but the FET takes up much less space.

In an alarm condition RA1 turns on relay Re1 via T1, which powers the bike’s indicators that are connected to K2. At each change in the alarm’s status (from standby to armed and vice versa) RA1 outputs a short signal and flashes the indicators twice (plus the number of times that the alarm has gone off). D2 and D3 suppress any possible induced voltages.

The power supply consists of just a 5 V regulator (IC2), which obtains its input voltage from the bike or scooter battery via connector K1. D1 prevents any damage from occurring to the circuit when the supply is connected with a reversed polarity. As mentioned earlier, two fuses have been used to improve the protection offered by the alarm. When an inventive thief thinks of opening an indicator and shorting the supply, F1 will blow but F2 will be unaffected. The circuit remains working in this case and the alarm is still noticeable because of the siren.

And now here is a reassuring statement regarding the supply: The whole circuit takes only 12 mA, both in standby as well as armed mode. So there is no need to worry that the battery will be discharged quickly.

Two circuits on one PCB

Since the alarm circuit and the remote control always complement one another, they are supplied as one PCB (**Figure 3**). This will have to be cut into two before use. That leaves you with two compact PCBs, with the transmitter having the approximate size of a small matchbox.

The transmitter PCB only contains the bare minimum of parts, so the construction shouldn’t take vary long. It’s only the SMD IC that requires accurate soldering and a soldering iron with a very fine tip. For beginners it is easiest to position the IC exactly and then keep it in place with a drop of glue. Then carefully solder one pin, check the position again and then solder the remaining pins. Excess solder can be removed later using desolder braid. Check carefully that no unwanted solder bridges remain between the IC pins! If you can’t get hold of the right battery holders for the button cells then you could always make your own simple frames from a few short lengths of thick tinned copper wire. That seems to work nearly as well. The button cells should be placed in their holders with the plus side facing upwards.

The construction of the main PCB shouldn’t give rise to any problems either. The artwork for the components shown in **Figure 3** is clear enough and the various connections are unambiguously marked. So as far as the ‘normal’ components are concerned, you shouldn’t have any problems. Most of the work for the main PCB will involve those parts that are not mounted directly on the board, but are external and are connected via short cables; this will be covered a little later.

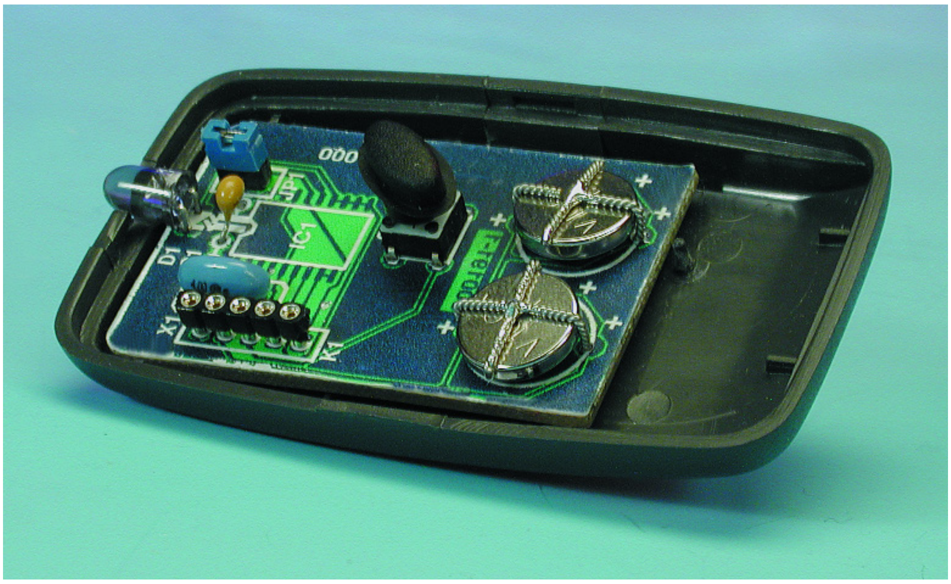

Figure 4. This small case is perfect for the transmitter PCB.

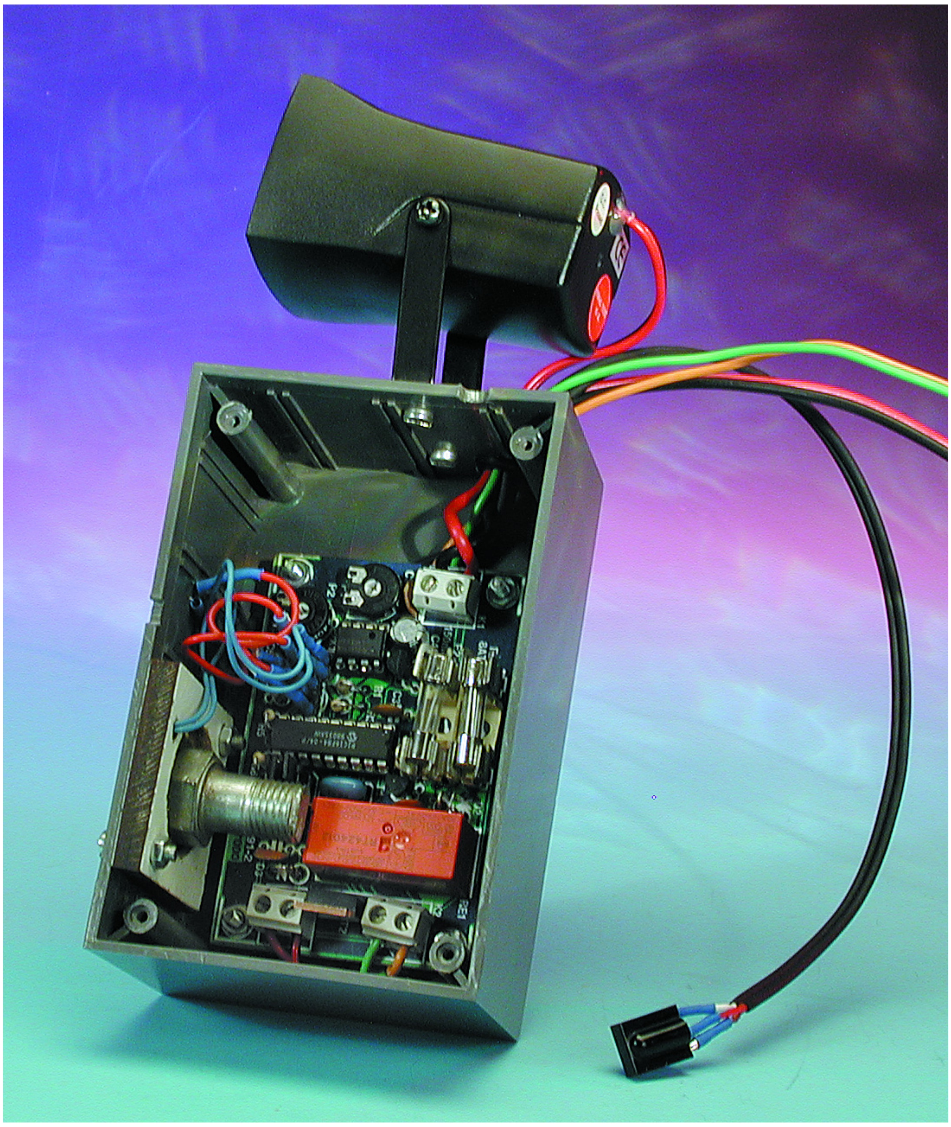

Figure 5. A view inside the prototype of the alarm. The way in which it is installed depends very much on the type of bike or scooter.

Installation

The transmitter PCB is so small that it could easily fit inside a small matchbox, including the button cells. There are also better and stronger cases that have been specifically designed for use as remote controls. For the prototype we used a case made by Teko, which can be obtained from Conrad (**Figure 4**). At the front you only need to make a small hole for D1. The box comes with a ready-made oval cutout along with a 'soft-touch' button that fits exactly above switch S1. Mounting the PCB into this box will barely take quarter of an hour.

The alarm PCB comes next. The way that is installed depends very much on the space available in the bike. Usually that can be found underneath the buddy seat, although the amount of space varies strongly from model to model. It could even be possible to leave out the case altogether and use a large piece of heat-shrinkable tubing round the PCB for insulation, and place that directly in the tool compartment of the bike. We have used a standard case to house our prototype (**Figure 5**) and we'd like to make a few important remarks regarding the construction and wiring.

It is obviously important to check that the polarity of the supply, IR receiver (IC4) and the siren are correct. Instead of the SFH5110-36 IR receiver used here, it is possible to use a SFH505; the only difference is that the connections are turned through 180° — so take care! When connecting the indicators to K2 you should take care that the left and right indicators remain isolated from each other, as otherwise they would both flash during normal operation.

IC4 should of course be mounted in such a way that the IR signal of the transmitter can reach it. A small hole will have to be made for this at the back of the bike; a slightly recessed position stops the IR receiver from 'seeing' signals that are not intended for it. A two-core shielded (audio) cable should be used to connect IC4 to the board, thereby minimising any possible interference.

The mercury tilt switches (S1 and

S2) should be mounted in such a way that they are closed when the bike or scooter is in its usual parking position (i.e., using the side or centre stand). It can be quite fiddly to find the best position for these.

LDR R2 has been mounted to the side of the case in our prototype. It is important that it is mounted in such a way that it can 'see the light' when the buddy seat is lifted, or the compartment that houses the alarm is opened.

This only leaves the mounting of the piezo transducer (X1).

Vibration sensor

The (cheap) piezo transducer used here turned out to be a very effective vibration sensor, so much so that in practice you could consider leaving out the mercury tilt switches altogether. It did however require a small modification. The transducer is glued to the side of the case (only glue along the edge, giving the active part the freedom to move!). A small weight of about 60 grams is then glued to the other side, pointing out sideways. The inertia of this weight increases the deformation of the piezo crystal that is caused by vibrations. This increases the sensitivity so much that this modification should be considered the most important part of the alarm. A 2.5 cm long M10 bolt seemed to work very well in practice.

If you want to keep everything tidy you could first make a plastic holder for the piezo transducer, using a drill and cutter. This can then be mounted to the alarm case using two screws (**Figure 6**). Since we had to experiment a fair amount on the prototype we decided on this practical solution.

Adjustments

Once the complete circuit has been constructed and the sensors, indicators and siren have been connected, you should first turn P1 and P2 fully counter clockwise before applying power from the battery via K1.

When the supply is connected for the first time the alarm system will be in its standby state. When the correct code from the remote control is received by the PIC16F84 and verified, the indicators will flash twice and the alarm will be armed. You can verify this state by looking at output RA0 of IC3: this should now be high.

P1 and P2 are then carefully turned clockwise until the required sensitivity has been set. Take care that you don't turn P2 too far because the piezo transducer can be made so sensitive that the alarm is almost triggered just by looking at the bike or scooter!

(000191-1)

Change the code?

In practice it is of course not very secure if all constructors of this alarm use the same infrared code to turn the alarm on and off. For those of you who know how to program, or have a friend who could help them, this can be rectified easily. You can easily change the fixed code into a personal one. For this you'll need an assembler and PIC programmer.

The source code files for both PIC controllers can be freely downloaded from the 'Free Downloads' section of our website (www.elektor-electronics.co.uk). The file is called **000191-11.zip** and the codes for the ready-programmed PICs are 000191-41 and 000191-42.

In both files the lines containing 'code 1' and 'code 2' should be searched for. The current values of 'F2' and '45' can then be changed to any value in the range '00' to 'FF'. Do make sure that both controllers have the same values! The modified programs are then assembled and finally burnt in the PIC controllers with a programmer.

A suitable DIY programmer for the PIC16F84 can be found on page 40 of the December 2000 issue of Elektor Electronics.

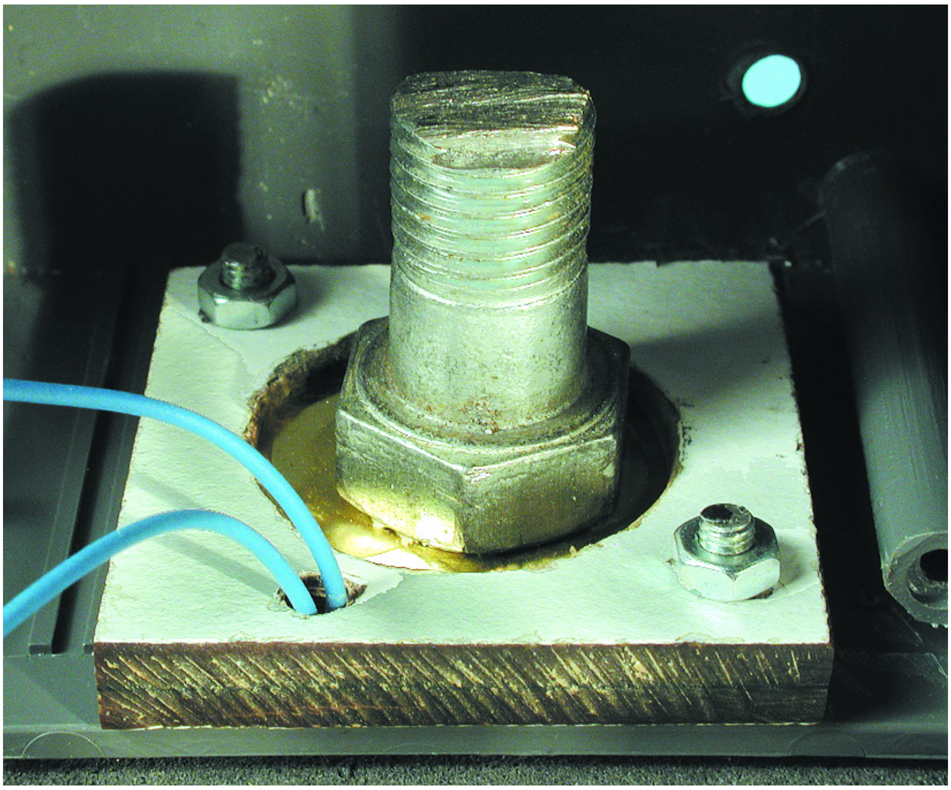

Figure 6. The piezo transducer is provided with the recommended counterweight, and then glued into a plastic frame.